Intelligent Customer Relationship Management on The Web

Kok Wai Wong, Chun Che Fung
Centre for Enterprise Collaboration in Innovative Systems
School of Information Technology, Murdoch University
Murdoch, Western Australia
[k.wong | l.fung] @murdoch.edu.au

Xiong Xiao
School of Computer Engineering, Nanyang Technological University
Singapore

Kit Po Wong
Department of Electrical Engineering
Hong Kong Polytechnic University
Hong Kong
eekpwong@polyu.edu.hk

***Abstract*— In recent years, Customer Relationship Management using web personalisation initiatives have gained much attention. The most important strategy of web personalisation is to provide the customers with correct information or services based on the knowledge about the customers' preferences. With the help of data mining technologies, the above strategy can be implemented. Computational intelligence technologies are investigated in this paper to provide interaction through web personalisation. This paper proposed two algorithms for the personalisation of the online shopping websites. It uses the Radial Basis Function (RBF) neural network. The algorithms first model the customer's preferences as a complex nonlinear function. It personalises the information presented to customers based on their preferences. The second is the preference learning algorithm. It learns the customer's preferences implicitly from the customer's behaviours by using a RBF neural network.**

I. Introduction

Since the 1990s, Internet has become popular for both businesses and personal usage. Due to its connectivity, Internet has become one of the largest "databases" in the world. It is also an important medium for business activities such as advertising, brand building, and online sales and services [1]. Traditional electronic commerce (e-commerce) usually relies on Value Added Networks (VAN) and private messaging networks, which are expensive and not widely accessible [2]. By harnessing the power of Internet, e-commerce is becoming cheaper, faster and more widely accessible. Various types of e-commerce such as B2B and B2C are growing at a tremendous speed.

Before the popular use of Internet, personalized business methods such as relationship marketing and target marketing are used to understand the needs and interests of customers [3]. Now with the use of e-commerce on Internet, personalization of business becomes important due to the excessive online product information and variety of customers' needs and interests. This excessive product information available normally makes it difficult for customers in making choices. The customers may also feel lost while browsing the large amount product information available and turn away from that business. Therefore, it is necessary to filter the information and present those that fit the customers' preferences. This process is now known as web personalization, which is an important component of Customer Relationship Management (CRM).

To implement web personalization, a customer's preference must be learnt from the past customers' profile. Considering that artificial neural networks are proven to have the ability to learn complex and unobvious rules from training patterns, it would be an interesting topic to apply neural network technology for the implementation of web personalization. Hence, in this paper, two personalization algorithms are proposed based on Radial Basis Functions (RBF) neural network. The first algorithm is used to filter product information and the second is to learn the customers' preference. The algorithms are then implemented and a simulator program is developed to test their effectiveness.

II. Web Personalisation

In a typical online shopping scenario with web personalization system [4] as shown in Figure 1, a customer first logs into the web site using his/her user name and password. The system will then retrieve the customer's profile from its storage (usually a database) and choose the contents to be presented based on the customer's profile. Eventually, the selected contents are assembled together to produce a customized web page and presented to the customer.

Data Collection

The first step to allow web personalization is to collect the customer's information. The information required usually depends on how the information will be used. Normally, these include the customer's personal particulars, preferences to products, the online behaviours, etc. There are several ways to collect the information. The two most commonly used methods are explicit profiling and inexplicit profiling [5].

Explicit profiling is done by asking the customer to provide information directly. It may be a web form, an interview, a survey or the customer's rating on certain products or services. Inexplicit profiling is the collection of data without customer's awareness, i.e. it is transparent to customers. This method includes the use of cookies, click-stream, etc [5]. The information that is collected is known as raw data. It usually needs to be pre-processed before being fed to the data analysis engines, such as data mining tools.

Before the collection of the customers' information, the website needs to convince the customers and gain permission in providing the information. The customers' privacy would be protected.

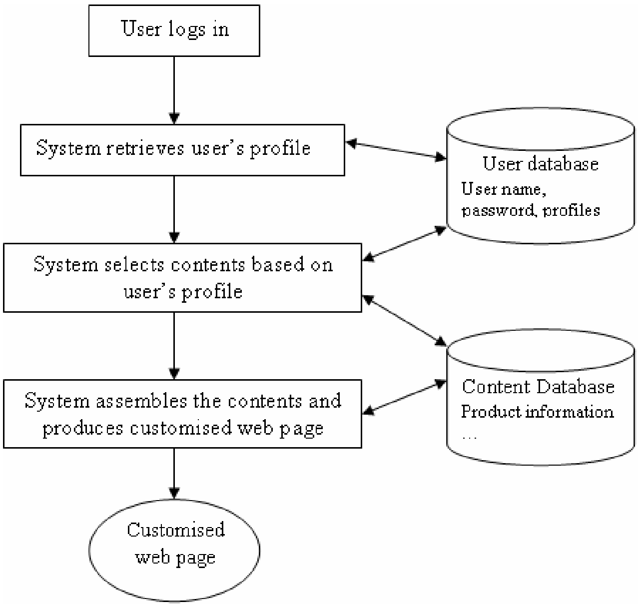


Fig. 1. Elements of Web Personalisation

Data Analysis

After the collection of the customers' information, several data analysis techniques (also known as filtering) can be applied to the data for providing recommendations to customers. There are four common data analysis techniques. They are rule-based analysis [5], simple filtering [4], content-based filtering [4], collaborative filtering [4], and hybrid filtering [3, 6]. Figure 2 shows the overview of a personalization system and the possible techniques used.

Web Personalization Technologies

In this section, the enabling technologies for web personalization will be discussed. The enabling technologies enable the personalization systems to do various kinds of data analysis, such as content-based filtering and collaborative filtering. Some common technologies include data mining, neural network, natural language generating, intelligent agents and information retrieval.

Data mining is a process aimed to find some rules or facts within a large amount of data without knowing the meaning of those data. It uses various kinds of technologies such as machine learning, statistical analysis, etc. There are four data mining techniques applicable to web personalization [5]. They are clustering, similarity indexing, association rules and classification.

Clustering refers to grouping a set of multidimensional points into several clusters so that the points in one cluster are close to each other. Clustering can be used for customer segmentation, pattern recognition and so on. Take its use in collaborative filtering as an example. A customer with n preferences can be represented as one n-dimensional vector or one point in n-dimension space. The preference may be on price, quality and so on. By clustering these vectors, customers with similar preferences are grouped into one cluster.

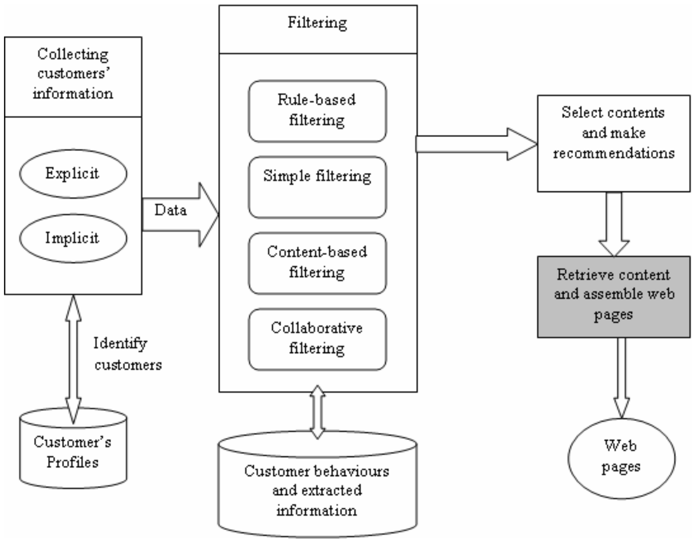


Fig. 2. Overview of the Personalisation Techniques

Similarity indexing is an approach to quickly identify people with similar taste by using an efficient index structure. It maps each person to a state vector based on his/her rating or purchase history.

Association rules technique is mainly used to find the buying patterns of customers. An association rule is a condition in the form A=>B, where A and B are two set of items. It implies that if the customer buys A, he/she will probably buy B as well. If a customer buys both A and B, we say this transaction is a support for the rule A=>B. And if we say the confidence of a rule A=>B is 80%, which implies 80% of customers, who buy A, also buy B. A useful rule must have high confidence. To find the rules, a large amount of transactions is analyzed. All combinations of the items are analyzed and only those combinations with high confidence are kept. They have the potential to become a rule, which may be used in rule-based filtering.

The classification problem is defined as follows. A set of training patterns is used to train the system. Each pattern has several attributes and a class tag. The attributes are input of the system and the class tag is the desired output of the system. These patterns are used to train the system. After successful training, the system is able to provide a correct class tag for the patterns when applying their attributes as input of the system. Such system can be used to predict the class tag of a testing pattern, which is not in the training patterns.

Neural network is a useful technology for web personalization. There are many types of neural networks, such as Multi-Layer Perceptron neural network (MLPNN), Radial Basis Functions (RBF) network. Yrjo and Mika [7] have proven that the Self-Organizing Map (SOM) network was able to cluster the customers according to their demographics and online behaviours. In their system, the customers' information, which was represented by an *n*-dimensional vector, was used to map the customers to clusters, which are in a two-dimensional discrete map. In the mapping process, no priori knowledge on the customers is required. These clusters are useful information for collaborative filtering. Many other kinds of neural networks can also be used to study the customer data and extract useful information. In this paper, a RBF network is used to learn the customer's preferences toward the products.

III. RBF PERSONALISATION SYSTEM

Nowadays, it is very common to buy goods online. When customers buy something, they usually want to compare several products to find the best choice. For example, if a customer wants to buy a Personal Digital Assistant (PDA), he/she may want the website to display several PDA models for choice. Without personalisation technology, the website had to give all the models to the customer. Sometimes the number of models is too many, and the customer needs a long time to view all of them in order to find the best match. Web personalisation can help to solve this problem by reducing the number of models displayed to customers. The idea is explained as follows. Suppose that the system knows the customers' preferences on PDA. Based on this knowledge about the customers, the system can filter out those PDA models that are unlikely to be chosen by the customers. Therefore, only the models with higher preference are displayed to customers. The number of displayed models can also be limited to a reasonable value comfortable to the customers. In this way, the customer is released from the excessive product information, which in turn makes them feel that the service is personal.

This paper focuses on the application of RBF neural network in personalising an online shopping system. The personalisation technique used is content-based filtering. The problem can be stated as follows. Suppose there are many products in an online shopping store. Every product has various attributes, such as price, quality, brand and guarantee. Different products have different values of attributes. This product information is stored in a database. The system also has the knowledge about the customers' preferences towards various kinds of products. This knowledge is learnt from customers' profiles. In this paper, the customers' profiles are the customers' rankings on products. Customers' preferences here include price preference, quality preference and so on. Each preference corresponds to one attribute of the products. When a customer wants to buy a product, the search engine first searches for all similar products in the product database. After that, based on the products attributes and customers' preferences, a filter is used to filter out those products that are unlikely to be selected by the customer, and the rest of the products are then presented to the customer. Figure 3 shows the flowchart of this e-commerce model.

In order to apply the content-based filtering on the products, the products need be modelled mathematically first. Different attributes may be quantified differently according to the nature of the attributes. Most of the attributes can be quantified easily. Every attribute of products is assumed to be quantified and normalized. A vector $A = \langle a_1, a_2, \cdots, a_n \rangle$ is used to represent the products mathematically, where *n* is the number of attributes.

Customers' profiles are necessary for the system to learn the customers' preferences. In this paper, assuming customers' ranking towards products is the only content in the profile. To get the ranking information, customer's online behaviours need to be collected and analyzed. The data structure [2] is used to store a customer's behaviour (see Fig. 4).

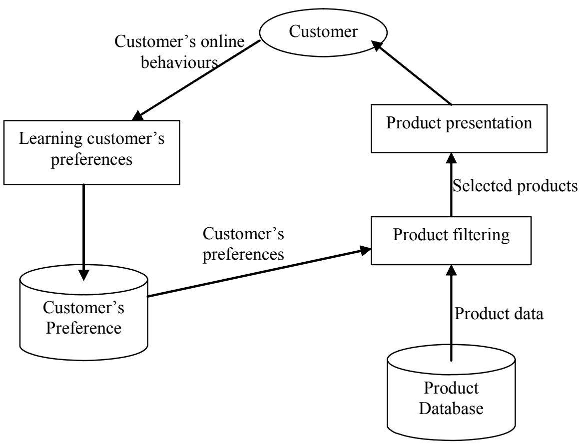
The item *display_times* is the number of times that the product is shown to the customer. A product may be shown to a customer multiple times during one session or in multiple sessions. The item *total_rank* is the accumulated customer's ranking towards the product through the history of the customer. Whenever a product is presented to a customer, the customer may choose it or ignore it. Based on the customer's behaviours, the *total_rank* is calculated in the following ways:

- When a new customer behaviour record is created, name is set to *customer's name* and *sku* is set to product's identifier. Both *display_times* and *total_rank* are initialized to zero.
- If the customer clicks the product to see its detailed description, the *total_rank* of the record increases by 1.
- If the customer adds the product to shopping basket, the *total_rank* increases by 2.
- If the customer purchases the product, the total_rank increases again by 2.

The above is a simple way of calculating *total_rank*. In real application, it may be modified to fit the different situation. After the customer's behaviours are collected, his/her ranking toward the product can be calculated using the formula:

$$Rank = total\_rank / display\_times$$

Once a customer's rankings on products are collected, a RBF neural network can be used to learn the customer's preferences. First, the customer's behaviours need to be modelled mathematically. This research proposed to model the customer's behaviours by nonlinear function. This is because the customer's behaviours can be viewed as a nonlinear function $t: X \rightarrow \Re$, which maps instances of a domain *X* to real values $\Re$. *X* is the domain of product attributes. The vectors used for representing products are in the domain *X*. The dependent variable of this function is the customer's rankings on products. The rankings are in the real value set $\Re$. By modelling the customer's behaviours using nonlinear function, learning of customer's preferences becomes a function approximation problem. As demonstrated by many applications, the RBF network is a good a function approximator, it is therefore used to approximate the customer's function. Each training pattern is a point on the surface of the function. The training of RBF network is aim to find a hypothesis $h: X \rightarrow \Re$. And the error of the hypothesis function is defined as $Q(x,h) = (t(x) - h(x))^2$ [8]. To get a good approximation, smaller error is desire.



.

Fig. 3. E-commerce model

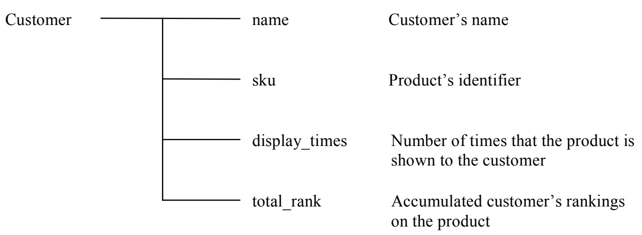


Fig. 4. The Data Structure of Customer's Behaviour

To model the customer's behaviours using a nonlinear function is the basic idea of this algorithm. The reason for this modelling is due to the similarity between the two. A customer reads in a product's information and extracts the product's attributes, and then gives a rank for the product. A function takes several independent variables and gives the value of the dependent variable.

In this part, the RBF neural network is the tool for approximating the customer's behaviours function. By carefully choosing a method to quantify the products' attributes and record the customer's behaviours, the proposed system should be able to learn the customer's preferences well and use it for prediction.

The filtering of the products is quite straightforward. First, the trained RBF network can be used to estimate the customer's ranking on the products. A higher ranking value implies a higher chance for the product to be preferred. The products are then sorted according to their ranking values. The first *n* highest-ranking products are presented to the customer, where *n* is the largest number of products that can be presented to customers.

IV. SIMULATION CASE

To verify the effectiveness of the system, a simulator is created. The purpose of the simulator is to test the effectiveness of the product filtering algorithm and the preference-learning algorithm. Therefore all customers' behaviours and the products attributes are simulated. A typical simulation consists of 20 sessions for one customer. In the simulation tests, only price and quality attributes are considered as a start.

The *ndpm* is a metric used widely for evaluating Information Retrieval systems [2]. It is used here to measure the effectiveness of the product filtering algorithm because of the similarity between the product filtering and the information retrieval. In the context of this research, *ndpm* is used to measure the difference between the customer's choices and the system's choices. Assume the symbol $\succ$ denotes a preference ranking, where $a \succ b$ means *a* is preferred to *b*. Let $\succ_c$ denotes customer ranking and $\succ_s$ denotes system ranking. Then the distance of the two rankings for a pair of product is calculated in the following way:

$$dist(a,b) = \begin{cases} 1 & if\,(a \succ_s b \; and \; b \succ_c a)\; or \;(b \succ_s a \; and \; a \succ_c b) \\ 0 & otherwise \end{cases} \tag{1}$$

The *ndpm* is the average distance of all pair of products.

$$ndpm = \frac{\sum_{a,b \in T} dist(a,b)}{|T|} \tag{2}$$

where *T* is the set of products and |*T*| is the number of pairs of the products.

The test results for independent product data without noise are shown in Fig. 5. The *x*-axis is the number of sessions passed, and the *y*-axis is the *ndpm*. The fallout is directly related to precision, so it is not shown. The legends of the curves specify the price and quality preferences of the customer and the weights of the preferences. For example, 1-0.5-5-0.5 means the price preference = 1, price weight = 0.5, quality preference = 5 and quality weight = 0.5. The summation of all the weights equals to one.

From Fig.5, it can be observed that the *ndpm* curves decrease slowly when the number of sessions increases. The decreasing rate is large in the beginning, and then it decreases slowly. The final values of the curves range from 1% to 1.5%. This can be considered as a good result. It is also noticed that the *ndpm* values for all curves are below 3.5% after the first session. This implies that the system can learn the customer's preferences quite accurately with little instances of the customer's behaviours. Although the overall trend of the curves is decreasing, there are some temporary increases in the curves. For the curve 3-0.5-3-0.5, the final value is not the minimum. This may be because the system is not stable enough.

Fig. 6 shows the test results on independent product data with different levels of noise. In the figure, the customer's preferences are set to <1,0.5,5,0.5>. This means that the price preference is 1 and quality preference is 5. And the weights for both of the two preferences are 0.5. This is believed to be a normal customer's preferences, as most customers want products to be cheap and high quality.

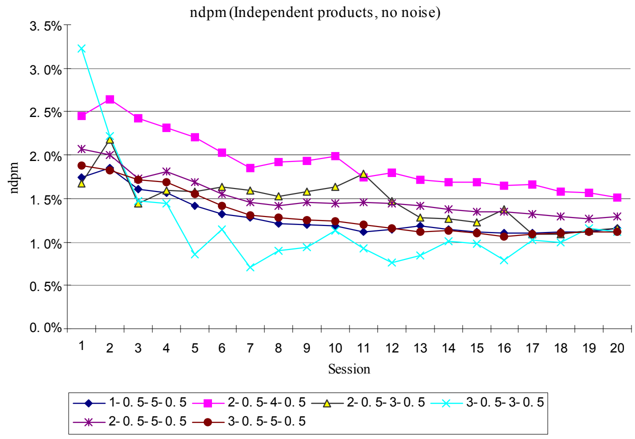


Fig. 5. The *ndpm* on Independent Products Without Noise

It can be shown in Fig. 6 that the *ndpm* increases as the *SNR* decreases. This implies that with a higher level of noise, the system's prediction becomes less accurate. This is easy to understand. As noise increases, the uncertainty of the customer's choice increases. Therefore, it becomes more difficult to predict the customer's behaviours. However, it is noticed that even when the *SNR*=10.5dB, the *ndpm* can be still reduced to below 15% finally. This implies that the system is quite resistant to noise.

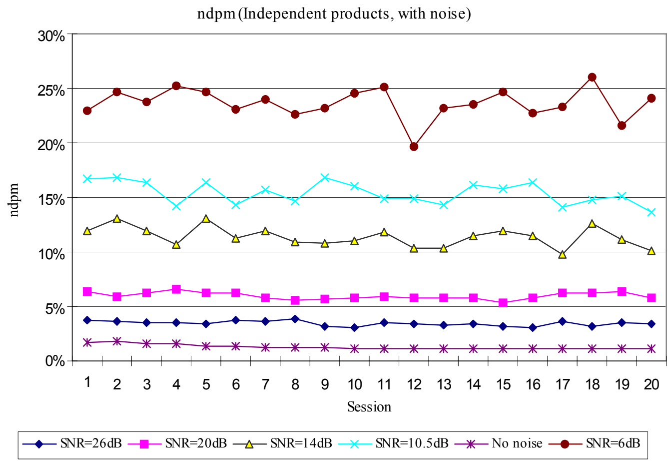


Fig. 6. The *ndpm* on Independent Products With Noise

V. CONCLUSION

This paper has shown two algorithms for the personalisation of the online shopping websites. It uses the RBF neural network. The algorithms first model the customer's preferences as a complex nonlinear function. It personalises the information presented to customers based on their preferences. The second is the preference learning algorithm. It learns the customer's preferences implicitly from the customer's behaviours by using a RBF neural network. Simulation results show that the proposed algorithm could be a good alternative in providing web personalisation based on collected customers profiles.

REFERENCES

[1] Hyperdictionary.com: Definition of Internet, Available: http://www.hyperdictionary.com/computing/internet

[2] Z. Wang, and C.K. Siew, "A Novel Approach to Personalization Product Filtering in Electronic Commerce," *Australian Journal of Intelligent Information Processing Systems,* v 7, n 3-4, 2001, pp. 81-95.

[3] S. Sae-Tang, and V. Esichaikul, *Web Personalization Techniques for E-commerce*, Active Media Technology, 2001, pp. 36-44.

[4] IBM high volume website team: Web site Personalization, 2000, Available: http://www-106.ibm.com/developerworks/websphere/library/techarticles/hvws/personalize.html.

[5] S. Murugesan, and A. Ramanathan, *Web Personalisation - An Overview,* Active Media Technology, 2001, pp. 65-76.

[6] M. Balabanovic and Y. Shoham, "Content-Based Collaborative Recommendation," *Communications of the ACM*, Vol. 40, No. 3.,1997.

[7] Y. Hiltunen, and M. Lappalainen, "Automated Personalization of Internet Users Using Self-Organizing Maps," *Proceedings of IDEAL,* 2002.

[8] MLnet Online Information Service: Definition of Function Approximation. Available at: http://kiew.cs.uni-dortmund.de:8001/mlnet/instances/81d91eaa-dac823c574.